

Proposition de correction de l'examen du cours *Systemes d'information*

Fabien Coelho et Claire Medrala – Mines Paris – PSL

Janvier 2022

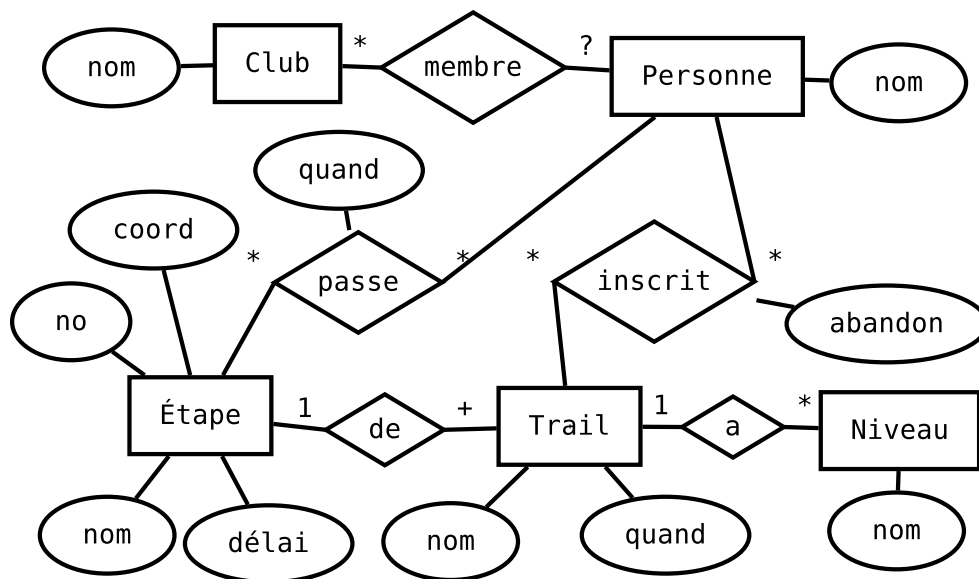
Bravo Hobbes ! 20/20

1 Modélisation entité-association

5/5

Un coureur à pied, éventuellement membre d'un club, peut s'inscrire à des trails (courses dans la nature) de différents niveaux. Un trail est composé d'étapes ordonnées. Une étape est un point géographique caractérisé par ses coordonnées et un délai maximal de passage pour ne pas être disqualifié de la course. La première et la dernière étape du trail constituent son départ et son arrivée. Lors d'une course, le système permet d'enregistrer l'heure de passage des coureurs à chaque étape. Un coureur peut abandonner sa course.

Voici une proposition de modèle E/A pour cette situation :



2 Traduction relationnelle

3/3

À partir du modèle E/A précédent, construisez un modèle relationnel.

```
CT Club(cid SPK, cnom TUNN);
CT Niveau(nid SPK, nmom TUNN);
CT Personne(pid SPK, pnom TUNN, cid IR Club);
CT Trail(tid SPK, tnom TNN, tqd DNN, nid INNR Niveau, U(tnom, tqd));
CT Inscrit(pid INNR Personne, tid INNR Trail, PK(pid, tid), abandon B);
CT Étape(eid SPK, eno INN, tid INNR Trail, U(enom, tid),
         enom TNN, ecoord PtNN, délai InNN, U(enim, tid));
CT Passe(pid INNR Personne, eid INNR Étape, PK(pid, eid), pqd TsNN);
```

3 Requêtes

8/8

On considère un modèle relationnel qui enregistre les locations d'hébergements de différents types dans différentes villes effectuées par des personnes.

1. Quelles sont les villes et adresses ordonnées des hébergements de type *maison* d'au moins 100 m² loués par *Calvin* (locataire) dans le département 971 ?

-- requête simple avec toutes les tables et conditions...

```
SELECT DISTINCT vnom, hadr
FROM Location
JOIN Personne USING (pid)
JOIN Hebergement USING (hid)
JOIN TypeHeb USING (tid)
JOIN Ville USING (vid)
WHERE tnom = 'maison'
      AND pnom = 'Calvin'
      AND vdept = 971
      AND hsurface >= 100
ORDER BY 1, 2;
```

Suggestions d'index :

-- À partir de tnom :

```
CREATE INDEX hebergement_tid ON Hebergement(tid);
CREATE INDEX location_hid ON Location(hid);
```

-- TypeHeb(tnom), Personne(pid), Ville(vid): existent déjà

-- À partir de pnom :

```
CREATE INDEX location_pid ON Location(pid);
```

-- Personne(pnom), Hebergement(hid), Ville(vid), TypeHeb(tid): existent déjà

-- À partir de vdept :

```
CREATE INDEX hebergement_vid ON Hebergement(vid);
```

-- Ville(vdept), TypeHeb(tid), Personne(pid): existent déjà

-- Location(hid): voir ci-dessus

-- À partir de hsurface:

```
CREATE INDEX hebergement_hsurface ON Hebergement(hsurface);
```

-- Ville(vid), TypeHeb(tid), Personne(pid): existent déjà

-- Location(hid): voir ci-dessus

2. Combien d'hébergements différents ont été loués en juillet 2020 (date d'arrivée) à un prix moyen par jour inférieur à 120 € ?

-- count distinct et calcul du prix par jour...

```
SELECT COUNT(DISTINCT hid)
FROM Hebergement
JOIN Location USING (hid)
WHERE lprix / (ldep - larr) <= 120.0
      AND larr BETWEEN '2020-07-01' AND '2020-07-31';
```

3. Quelles sont les noms ordonnés des propriétaires qui n'ont jamais été locataires ?

```
-- avec une différence
SELECT pnom
FROM Personne
JOIN Hebergement USING (pid)
EXCEPT
SELECT pnom
FROM Personne
JOIN Location USING (pid)
ORDER BY 1;
```

```
-- avec une jointure externe
SELECT DISTINCT pnom
FROM Personne
JOIN Hebergement USING (pid)
LEFT JOIN Location USING (pid)
WHERE lid IS NULL
ORDER BY 1;
```

4. Quels sont les logements ordonnés (département, ville, adresse) avec leur durée moyenne de location et leur nombre total de locations, pour lesquels la durée moyenne de location est supérieure à la moyenne globale des durées de toutes les locations ?

```
-- calcul de durée et having avec une sous-requête
SELECT vdept, vnom, hadr, AVG(ldep - larr) AS durée, COUNT(*) AS nloc
FROM Hebergement
JOIN Ville USING (vid)
JOIN Location USING (hid)
GROUP BY 1, 2, 3
HAVING AVG(ldep-larr) >= (SELECT AVG(ldep-larr) FROM Location)
ORDER BY 1, 2, 3
```

5. Quels sont les noms ordonnés des propriétaire qui ont loué leurs biens plusieurs fois à eux-mêmes ?

```
-- jointure sur deux conditions, having simple sur un compte
SELECT pnom
FROM Personne
JOIN Hebergement USING (pid)
JOIN Location USING (hid, pid)
GROUP BY 1
HAVING COUNT(*) > 1
ORDER BY 1;
```

6. Pour tous les hébergements ordonnés (département, ville, adresse), quel est le nombre (y compris 0) de locations commencées dans l'année 2020 et celles commencées dans l'année 2021 ?

```

-- jointure gauche et comptages avec des filtres
SELECT
    vdept, vnom, hadr,
    COUNT(lid) FILTER(WHERE EXTRACT(YEAR FROM larr) = 2020) AS "2020",
    COUNT(lid) FILTER(WHERE EXTRACT(YEAR FROM larr) = 2021) AS "2021"
FROM Hebergement
JOIN Ville USING (vid)
LEFT JOIN Location USING (hid)
GROUP BY 1, 2, 3
ORDER BY 1, 2, 3;

```

7. Quels hébergements ordonnés (département, ville, adresse) ont été loués *en continu* (locations successives sans interruptions) pendant au moins 300 jours ?

```

-- récursion pour les locations continues
WITH RECURSIVE Continu(hid, larr, ldep) AS (
    SELECT hid, larr, ldep
    FROM Location
    UNION
    SELECT c.hid, c.larr, l.ldep
    FROM Continu AS c
    JOIN Location AS l ON (c.ldep = l.larr AND c.hid = l.hid)
    -- pas de condition d'arrêt car pas de cycle possible
)
SELECT DISTINCT vdept, vnom, hadr
FROM Continu AS c
JOIN Hebergement USING (hid)
JOIN Ville USING (vid)
WHERE c.ldep - c.larr >= 300
ORDER BY 1, 2, 3;

```

4 Questions de cours

4/4

Choisissez un thème parmi les deux tirés aléatoirement en début d'examen dans la liste *Postgres, Relationnel, Optimisation, Droits, Transaction, MVCC, PL/pgSQL, REST, Python DB API, Injection SQL, Décisionnel, Formes normales* et expliquez en moins de 100 mots ce que vous en avez retenu.

Thème : ACID (Advanced Chanting In Databases)

Ce thème est très passionnant, j'ai appris plein de choses super intéressantes qui éclairent parfaitement le fonctionnement de cet aspect des bases de données à la fois théorique et pratique, et me permettent de bien comprendre les caractéristiques générales et particulières des multiples facettes de ce thème dans une perspective transversale de mise en application concrète du modèle relationnel sur des problèmes réels, tout en gardant, au delà du simple niveau fonctionnel, une maîtrise généraliste des aspects opérationnels sur Postgres qui gère le stockage à un prix compatible avec toutes les bourses, une bonne nouvelle pour nos budgets !

Citez les noms de deux scientifiques ayant obtenu le prix Turing pour leurs travaux de recherche sur les bases de données.

Il y a 4 prix Turing liés aux recherches sur les bases de données : *Charles Bachman, Edgard Codd, James Gray* et *Michael Stonebraker*.

Document composé avec L^AT_EX.